CLAIMS

What is claimed is:

1. A method of outputting a first TCP/IP packet and a second TCP/IP packet from a network interface device, the first TCP/IP packet and the second TCP/IP packet being output to a network, comprising:
(a) storing first packet information on the network interface device;
(b) pushing a first pointer to the first packet information onto a first transmit queue of the network interface device;
(c) storing second packet information on the network interface device;
(d) pushing a second pointer to the second packet information onto a second transmit queue of the network interface device; and
(e) popping the second pointer off the second transmit queue and then popping the first pointer off the first transmit queue, the popped second pointer being used to locate the second packet information, the located second packet information then being output from the network interface device in the form of a second TCP/IP packet, the popped first pointer being used to locate the first packet information, the located first packet information being output from the network interface device in the form of a first TCP/IP packet such that the second TCP/IP packet is output from the network interface device and to the network before the first TCP/IP packet is output from the network interface device and to the network.

2. The method of Claim 1, wherein the first TCP/IP packet is a data packet, wherein the second TCP/IP packet is a control packet, and wherein the network interface device is coupled to a host computer by a parallel bus.

3. The method of Claim 1, wherein the first transmit queue contains pointers associated with a first set of packets, and wherein the second transmit queue contains pointers associated with a second set of packets, the second set of packets having transmission priority over the first set of packets.

4. The method of Claim 1, wherein the network interface device comprises a transmit sequencer, a memory, and MAC interface circuitry, the transmit sequencer causing the second packet information to be transferred from the memory to the MAC interface circuitry, the second TCP/IP packet being output from the network interface device through the MAC interface circuitry.

5. The method of Claim 1, wherein the first TCP/IP packet is associated with a first TCP/IP connection, and wherein the second TCP/IP packet is associated with a second TCP/IP connection, and wherein the pushing of (b) occurs before the pushing of (d).

6. The method of Claim 1, wherein the first packet information includes a header portion and a data payload portion.

7. The method of Claim 1, wherein the first pointer is part of a buffer descriptor.

8. The method of Claim 1, further comprising:
receiving onto the network interface device from the network a third packet;
fast-path processing the third packet on the network interface device such that a data payload portion of the third packet is written into a destination memory without a network protocol stack performing substantial transport or substantial network layer protocol processing on the third packet;
receiving onto the network interface device from the network a fourth packet; and
slow-path processing the fourth packet such that at least a data payload portion of the fourth packet is written into the destination memory, the network protocol stack performing substantial transport and substantial network layer protocol processing on the fourth packet.

9. The method of Claim 1, wherein the second TCP/IP packet is a TCP ACK.

10. The method of Claim 8, wherein the second TCP/IP packet is a TCP ACK.

11. The method of Claim 1, wherein the second transmit queue is used for the transmission of TCP ACKs, and wherein the first transmit queue is used for the transmission of TCP/IP data packets, the second transmit queue being free of or substantially free of pointers to TCP/IP data packets.

12. The method of Claim 8, wherein the network protocol stack is executed by a processor, the processor being a part of the network interface device.

13. The method of Claim 8, wherein the network protocol stack is executed by a processor, the processor being a part of a host computer, the network interface device being coupled to the host computer.

14. A TCP/IP offload network interface device, comprising:
a memory containing first packet information and second packet information;
a processor that causes a first pointer to the first packet information to be pushed onto a first transmit queue before a second pointer to the second packet information is pushed onto a second transmit queue; and
a transmit mechanism that pops the second queue in preference to popping the first queue, the transmit mechanism popping the second pointer off the second queue and outputting the second packet information from the network interface device in the form of a second TCP/IP packet, the transmit mechanism popping the first pointer off the first queue and outputting the first packet information from the network interface device in the form of a first TCP/IP packet, the transmit mechanism popping the second pointer from the second queue before popping the first pointer off the first queue, the second TCP/IP packet being output from the network interface device before the first TCP/IP packet is output from the network interface device.

15. The TCP/IP offload network interface device of Claim 14, wherein the first TCP/IP packet is a data packet, and wherein the second TCP/IP packet is a control packet.

16. The TCP/IP offload network interface device of Claim 14, wherein the first TCP/IP packet is a data packet associated with a first TCP/IP connection, and wherein the second TCP/IP packet is a TCP ACK associated with a second TCP/IP connection.

17. The TCP/IP offload network interface device of Claim 16, wherein the TCP/IP offload network interface device is operatively coupled to a host computer, the host computer executing a protocol processing stack.

18. The TCP/IP offload network interface device of Claim 16, wherein the network interface device includes a second processor, the second processor executing a protocol processing stack, the second processor being part of the TCP/IP offload network interface device.

19. The TCP/IP offload network interface device of Claim 16, wherein a template header having TCP fields and IP fields is stored on the TCP/IP offload network interface device, and wherein the processor executes a transmit finite state machine, the transmit finite state machine filling in the TCP fields and the IP fields of the template header, the filled in template header forming at least a part of the second packet information.

20. The TCP/IP offload network interface device of Claim 19, wherein the transmit finite state machine does not include a TCP protocol processing layer and a discrete IP protocol processing layer, but rather the transmit finite state machine covers both TCP and IP protocol processing.

21. A method for outputting an acknowledge (ACK) from a protocol processing offload network interface device (PPONID), the PPONID being coupled to a network, the method comprising:
receiving a first packet onto the PPONID from the network;
slow-path processing the first packet such that a protocol processing stack performs substantial transport layer processing and substantial network layer processing on the first packet;
receiving a second packet onto the PPONID from the network;

fast-path processing the second packet on the PPONID such that the stack performs substantially no transport layer processing on the second packet and such that the stack performs substantially no network layer processing on the second packet;
pushing a first pointer to first packet information onto a first transmit queue;
in response to said receiving of the second packet pushing a second pointer to second packet information onto a second transmit queue;
popping the second transmit queue to retrieve the second pointer, and using the second pointer to retrieve the second packet information, and outputting the second packet information from the PPONID in the form of the ACK; and
popping the first transmit queue to retrieve the first pointer, and using the first pointer to retrieve the first packet information, and outputting the first packet information from the PPONID in the form of a third packet, the ACK being output from the PPONID before the third packet.

22. The method of Claim 21, wherein PPONID is coupled to a host computer, and wherein the second packet includes a data payload, the data payload being transferred from the PPONID and to the host, the ACK being output from the PPONID before any portion of the data payload is transferred to the host.

23. A method for outputting a TCP ACK from a network interface device, the network interface device being coupled to a network, the method comprising:
receiving a first TCP/IP packet onto the network interface device from the network;
slow-path processing the first TCP/IP packet such that a protocol processing stack performs substantial TCP layer processing and substantial IP layer processing on the first TCP/IP packet;
receiving a second TCP/IP packet onto the network interface device from the network;
fast-path processing the second TCP/IP packet on the network interface device such that the stack performs substantially no TCP layer processing on the second TCP/IP packet and such that the stack performs substantially no IP layer processing on the second TCP/IP packet;
pushing a first pointer to third packet information onto a first transmit queue;

in response to said receiving of the second TCP/IP packet pushing a second pointer to fourth packet information onto a second transmit queue;
popping the second transmit queue to retrieve the second pointer, and using the second pointer to retrieve the fourth packet information, and outputting the fourth packet information from the network interface device in the form of the TCP ACK; and
popping the first transmit queue to retrieve the first pointer, and using the first pointer to retrieve the third packet information, and outputting the third packet information from the network interface device in the form of a third TCP/IP packet, the TCP ACK being output from the network interface device before the third TCP/IP packet.

24. The method of Claim 23, wherein the protocol processing stack executes on a host computer, the network interface device being coupled to the host computer.

25. The method of Claim 23, wherein the protocol processing stack executes on a processor, the processor being part of the network interface device.

26. The method of Claim 23, wherein the network interface device comprises a processor, the method further comprising the step of generating the fourth packet information, the step of generating the fourth packet information comprising:
accessing a template header stored on the network interface device, the template header having TCP fields and IP fields; and
executing a finite state machine on the processor, the finite state machine filling in the TCP fields and the IP fields and thereby forming the fourth packet information without passing the fourth packet information or the template header down to any lower protocol processing layer in any protocol stack.

27. The method of Claim 26, wherein the network interface device further comprises a transmit sequencer, the transmit sequencer causing the TCP ACK and the third TCP/IP packet to be output from the network interface device.

28. A method of generating an ACK on a TCP/IP offload device, comprising:

(a) step for using a template header to generate the ACK without sequentially processing the template header through a TCP protocol processing layer on the TCP/IP offload device and then processing the template header through an IP protocol processing layer on the TCP/IP offload device, the template header having TCP and IP fields, step (a) involving filling in the TCP and IP fields;

(b) pushing pointers to data packet information onto a first transmit queue, and then pushing a pointer to the filled-in template header onto a second transmit queue; and

(c) popping the second transmit queue in preference to popping the first transmit queue such that the second transmit queue has transmission priority over the first transmit queue, and outputting from the TCP/IP offload device the filled-in template header in the form of the ACK, the data packet information being output from the TCP/IP offload device in the form of data packets, the ACK being output from the TCP/IP offload device before any of the data packets are output from the TCP/IP offload device.